

Adabas D And Other SQL Systems

This section covers the following topics:

- General Information
 - Documenting SQL Systems in Predict
 - Naming Conventions for SQL Objects
 - Generating SQL CREATE Statements
 - Generating DDMs from SQL Objects
 - Incorporating Tables / Views of SQL Database Systems
 - Administrating SQL Objects
-

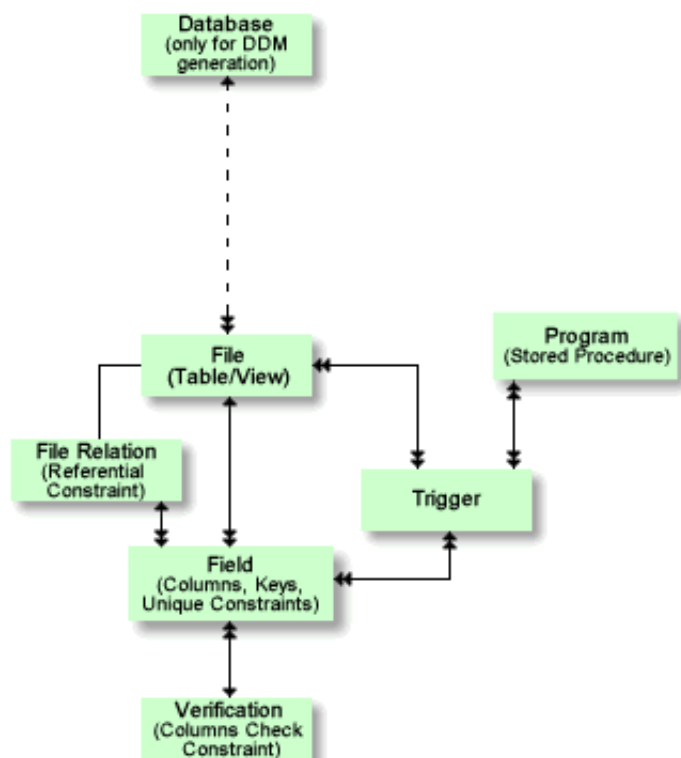
General Information

Predict offers enhanced support for the following SQL systems:

- Adabas D
- Oracle
- Ingres
- Informix
- Sybase

The following SQL objects and attributes can be documented in Predict. Not all attributes are applicable for all SQL systems.

- Tables and Views
- Columns
- Referential constraints
- Unique constraints
- Check constraints
- Stored procedures
- Triggers



Documenting SQL Systems in Predict

Documenting SQL Tables and Views

The following table gives an overview of how different SQL objects are documented in Predict.

SQL Object	Documented in Predict with File of Type
Adabas D Table, View	BT, BV
Oracle Table, View	OT, OV
Ingres Table, View	JT, JV
Informix Table, View	XT, XV
Sybase Table, View	YT, YV
Other SQL systems	X

Subselect Clauses and Derived Field Expressions

The documentation of views is supported by the Natural-based Subquery Editor in Predict to specify the part of the subselect clause starting from the first FROM clause.

The selection clause of the subselect clause is documented by the specification of the field list of the view. The specified list of tables/views in the first FROM clause of the subselect clause is generated by Predict and will be updated if a field from an additional table/view is added to the view. Correlation names can be added to the tables

and views in the list (using editor functions). The remaining part of the subselect clause is left unchanged.

The expression used to define SQL fields can contain complex expressions. fields that are defined not only by a single column name but use either a constant or a more complex expression are called derived fields. These derived fields can be edited with the Subquery Editor.

Comment lines (starting with * or **) and remarks within a line (starting with /*) are allowed in derived field expressions and subselect clauses.

Documenting Other SQL Objects

SQL Object	Valid for					Documented in Predict with Object of Type	Note
	BT,BV	OT,OV	JT,JV	XT,XV	YT,YV		
Trigger			Y	Y	Y	Trigger	'Triggers' are referred to as 'Rules' in Ingres
Table Check Constraint	Y	Y	Y	Y		Attribute of file	
Column Check Constraint	Y	Y	Y	Y	Y	Verification of status SQL	A 'Column check constraint' is referred to as 'Integrity' in Ingres and 'Rule' in Sybase.
Stored Procedure	(Y)	(Y)	Y	Y	Y	Program of type R	
Column	Y	Y	Y	Y	Y	Field	
Common Key					Y	Field	
Primary Key	Y	Y	Y	Y	Y	Field	
Foreign Key	Y	Y	Y	Y	Y	Field	
Unique constraint	Y	Y	Y	Y	Y	Field	
Referential constraint	Y	Y	Y	Y	Y	File Relation of type R	

Note:

Objects marked with (Y) can be documented in Predict but are not included in the respective CREATE statement.

SQL Columns

SQL columns are documented as field objects.

Keys

SQL keys are documented as follows:

- If the key consists of only **one** column, the field documenting the column is marked as a descriptor with descriptor type P, E or K.
- If the key applies to **multiple** columns, it is documented as a field of type SP (superfield) with descriptor type P, E or K. The descriptor types have the following meaning:
 - **E**
Field is a foreign key.

- **K**
Field is a common key.
- **P**
Field is a primary key. This always implies that the field has a unique constraint.

Unique Constraints

Unique constraints are documented as follows:

- If the unique constrain applies to only **one** column, the field documenting the column is marked U in column Unique option.
- If the unique constraint applies to **multiple** columns, it is documented as a field of type SP (superfield) with descriptor type P, E or K, and U in column Unique option. The descriptor types have the following meaning:
 - **E**
Field is a unique foreign key.
 - **K**
Field is a unique common key.
 - **P**
Field is a primary key. This always implies that the field has a unique constraint.

Common Keys

Common keys (columns that are frequently joined between two tables or views) are documented in Predict with a file relation of type K. The two fields for which the relationship is to be established must have descriptor type K.

Common keys are only applicable to Sybase.

Referential Constraints

Referential constraints are documented as file relations of type R (referential constraint). A relationship is established between a unique key and a foreign key. Unique and foreign key can belong to the same or to different tables.

Column Check Expressions

Check expressions for single columns are documented with verifications of status SQL. The check expression is stored as the rule of the Verification.

Check expressions can be edited with the Predict Rule Editor.

Comment lines (lines starting with * or **) and remarks within a line (starting with /*) are allowed.

Table Check Expressions

A table check expression is a check expression that applies to more than one column. A table check expression is an attribute of a file.

To edit table check expressions, enter Y in the field Trigger of the corresponding file object.

Comment lines (lines starting with * or **) and remarks within a line (starting with /*) are allowed.

Triggers

See the section Trigger in the **Predefined Object Types in Predict** documentation

Stored Procedure

Stored procedures are documented the procedure code of programs of type R (SQL procedure) and language S (SQL).

If the trigger of a file of type XT, YT or JT contains the text EXECUTE procedure_name, and the procedure_name corresponds to a program of type R and language Q, then the procedure code of the program object is included in the generated CREATE statement.

Naming Conventions for SQL Objects

Special naming conventions apply to the following objects in Predict

- SQL file types. See table below.
- Fields linked as children to these file types
- Constraint names
- Correlation names
- Tablespace for Oracle
- Procedure/Function name

The file IDs must be fully qualified.

A fully qualified ID consists of three parts:

- Creator of up to 8 characters
- Hyphen to separate creator from table/view name
- Table/view name. The maximum length depends on the SQL system. See table below.

Fully qualified IDs may not exceed 32 characters. For SQL objects where the table/view name may not exceed 18 characters, the maximum length of the fully qualified ID in Predict is 27.

The permitted characters listed in the table below apply to creator and table/view name.

File Type	BT, BV	JT, JV	OT, OV	XT, XV	YT, YV
Convention					
Maximum length of table/view name	18	24	30	18	30
Upper case			Y		
Upper/lower case	Y	Y		Y	Y
'_' allowed at first pos.		Y			Y
'#' allowed at first pos.	Y				
'\$' allowed at first pos.	Y				
'@' allowed at first pos.	Y				
'_' allowed from second position	Y	Y	Y	Y	Y
'#' allowed from second position	Y	Y	Y		Y
'\$' allowed from second position	Y	Y	Y		Y
'@' allowed from sec. position	Y	Y			Y
Numbers allowed from second position	Y	Y	Y	Y	Y

Correlation Names

Correlation names can be defined in the subselect clause of a view. If a correlation name is defined for a table/view in the subselect clause, all references (in field expressions as well as in the field editor of the view) to columns of the table/view must be qualified with the correlation name. If no correlation name is defined for a table/view in the subselect clause, all references to columns of the table/view must be fully qualified with creator-tablename-columnname (for example: SYSIBM-SYSCOLUMNS-COLNAME).

Delimited Identifiers

It is possible to incorporate SQL tables and views that have delimited identifiers. These tables and views can then be renamed with Predict maintenance functions, and retrieval functions can be applied to them. It is strongly advisable to rename delimited identifiers for the following reasons:

- The only Predict functions that can be applied without restriction to objects with delimited identifiers are Incorporate and Rename.
- If identifiers contain special characters such as blank or asterisk, results of retrieval functions are unpredictable.
- Views can only be generated if the subselect clause and the column expressions do not contain references to delimited identifiers enclosed by quotation marks.

Generating SQL CREATE Statements

Functional Scope

The following table gives an overview of the CREATE statements that can be generated from Predict objects with the function Generate SQL CREATE Statement. These SQL statements are stored as Natural members.

Note:

If a CREATE statement is not available for a particular SQL system, a corresponding clause is generated in the CREATE TABLE or CREATE VIEW statement if applicable.

CREATE STATEMENT	File Type						
	BT,BV	D,E	OT,OV	JT, JV	X	XT,XV	YT,YV
TABLE, VIEW	Y	Y	Y	Y	Y	Y	Y
INDEX		Y					
DEFAULT						Y	Y
RULE				Y		Y	Y
PROCEDURE				Y		Y	Y
TRIGGER				Y		Y	Y
LABEL ON		Y					
COMMENT ON		Y	Y				

The statements can be punched to an operating system member for further processing, for example execution with an interactive SQL tool or a user program.

More Information

For more information see the section Generation in the **External Objects in Predict documentation**.

Generating DDMs from SQL Objects

The following rules apply when generating a DDM for Natural from an SQL file object:

- The file must be linked via *Contains FI* to a database of a compatible type:

File Type		Compatible Database Type	
BT, BV	Adabas D Table, View	B	Adabas D Handler
JT, JV	Ingres Table, View	J	Ingres Handler
OT, OV	Oracle Table, View	O	Oracle Handler
XT, XV	Informix Table, View	X	Informix Handler
YT, YV	Sybase Table, View	Y	Sybase Handler

- Files documenting both tables and views must be linked to a database of the corresponding type.
- The file must be linked via *Contains FI* to a database of which the database number is defined in the NATCONF.CFG file of type OSQ.

For more information see Data Definition Module in the section Generation in the **External Objects in Predict documentation**.

Incorporating Tables / Views of SQL Database Systems

Incorporation of tables and views of SQL systems is subject to security mechanisms of the respective system.

Tables and views of the following SQL systems can be incorporated.

- Adabas D
- Ingres
- Informix
- Oracle
- Sybase

Required Access

Access is required to the following in the respective SQL system:

Adabas D

- Command SHOW

Ingres

- IICOLUMNS
- IIDBDEPENDS
- IIINTEGRITIES
- IIRELATION
- IITABLES
- IIVIEWS

Informix

- SYSCOLUMNS
- SYSCONSTRAINTS
- SYSDEPEND
- SYSINDEXES
- SYSTABLES
- SYSUSERS
- SYSVIEWS

Oracle

- SYS.DBA_CATALOG
- SYS.DBA_COL_COMMENTS
- SYS.DBA_CONS_COLUMNS
- SYS.DBA_CROSS_REFS
- SYS.DBA_TAB_COLUMNS
- SYS.DBA_VIEWS
- ALL_CATALOG
- ALL_COL_COMMENTS
- ALL_CONS_COLUMNS
- ALL_CROSS_REFS
- ALL_TAB_COLUMNS
- ALL_VIEWS

Sybase

- master.dbo.spt_values
- syscolumns
- syscomments
- sysdepends
- syskeys
- sysobjects
- systypes

More Information

For more information see the section Incorporation in the **External Objects in Predict documentation**.

Administrating SQL Objects

You can display and purge the generation protocols created by Predict from the function Generate SQL CREATE statement. You cannot process objects in the external SQL environment with administration functions.

More Information

For more information see the section Administration of External Objects in the **External Objects in Predict documentation**.